Appendix F

Example Script to Pick Up Acks and Send a File

EXAMPLE SCRIPT TO PICK UP ACKS AND SEND A FILE

The following example is meant to illustrate how a script can communicate with EMS using only pattern-matching to control logic flow. The example is for reference only, and is not intended for actual use by trading partners.

This particular script is written in "expect", designed for a Unix operating system, and takes advantage of expect's ability to specify a set of strings and events to be watched for if there is a failure to match the string that is anticipated. It also assumes that files are to be sent and received using the FTP protocol. A script that instead used the Zmodem protocol, for example, would differ noticeably in those parts of the script that accomplish the actual sending and receiving of files.

Most "expect" commands and syntax appearing in the script are reasonably intuitive, at least for the limited purposes of illustration for which this script is intended. But it is worth mentioning that the command "send" directs output to the telnet session, while the command "send_user" directs output to "standard output", which is assumed to be directed to a local log file. Lines beginning with "#" are comments.

```
#!/opt/sfw/bin/expect  -f $1 $2 $3 $4 $5 $6

#Assign command-line parameters to local variables for convenience.
set log_id  [lrange $argv 0 0]
set passwd  [lrange $argv 1 1]
set hostid  [lrange $argv 2 2]
set retfil  [lrange $argv 3 3]
set prtocl  [lrange $argv 4 4]
set compid  [lrange $argv 5 5]


# Slow down "typing" of replies to allow for modem turnaround delays.
set send_slow {1 .1}

# Start a C-shell in which to run telnet
spawn /usr/bin/csh

# Specify set of "secondary" strings/events to be watched for if anticipated match fails.
# These messages and events could occur at any time during processing.
#   message: "EFS is down"
#   message: "DISCONNECTING FROM EFS"
#   event:   eof (telnet session was terminated for any reason, e.g., EMS disconnects)
expect_after {

-exact "     EFS is busy. Wait at least 10 minutes, then retry." {
               send_user "got the EFS BUSY message (abort) \n"
               exit }

-exact "     EFS is currently unavailable.  Additional information may be available on
IRS quick alerts. " {
               send_user "got the EFS UNAVAILABLE message (abort) \n"
               exit }

-exact "EFS DISCONNECTINg FROM EFS" {
               send_user "got the DISCONNECTING message (abort) \n"
               exit }

       eof   {
               send_user "tp_client disconnected (abort) \n"
               exit }
```

```
}

# When C-shell prompt appears,
#   Start a telnet session to the designated computer (hostid)
#   Exit the C-shell when the telnet session exits (even if that
#   occurs before the script runs to completion)
expect -exact "% "
sleep .1
send -s -- "telnet $hostid; exit\r"

# When login prompt from EMS is received, send username (log_id).
expect -exact "login: "
sleep .2
send -s -- "$log_id\r"

#When password prompt from EMS is received, send password (passwd)
expect -exact "Password:"
sleep .2
send -s -- "$passwd\r"

#When MAIN MENU choice-prompt from EMS is received,
#send 3 (Change File Transfer Protocol)
expect -exact "    Enter your choice: "
sleep .2
send -s -- "3\r"

#When FILE TRANSFERS PROTOTCOL MENU choice-prompt from EMS is received,
#send protocol to use (prtocl)
expect -exact "    Enter your choice: "
sleep .2
send -s -- "$prtocl\r"

#When MAIN MENU choice-prompt from EMS is received,
#send 4 (Change Compression Method)
expect -exact "    Enter your choice: "
sleep .2
send -s -- "4\r"

#When COMPRESSION METHODS MENU choice-prompt from EMS is received,
#send compression to use (compid)
expect -exact "    Enter your choice: "
sleep .2
send -s -- "$compid\r"

#When MAIN MENU choice-prompt from EMS is received,
#send 2 (Receive/Send File(s))
expect -exact "    Enter your choice: "
sleep .2
send -s -- "2\r"

#If there are acks to pick up, EMS will prompt for the TP to receive them.
#If not, or after they have been picked up, EMS will prompt to allow sending a file.
#The logic below handles both possibilities.
#If there are files to pick up, the logic responds "y" to receive them.
#After they are received, it responds "y" to the prompt for sending a file,
#then responds with the local filename to be sent, because this script assumes
#that the FTP protocol is being used.
#If there are not any files to pick up, the logic responds "y" to the prompt for sending
#a file, then responds with the local filename to be sent, because this script assumes
#that the FTP protocol is being used.
expect {
-exact  "    Are you ready to receive files? Y/\[N\]: " {
                  sleep .2
                  send -s -- "y\r"
                  expect -exact  "    Do you want to send a file? Y/\[N\]: "
                  sleep .2
                  send -s -- "y\r"
                  expect -exact "    are sending from your system: "
                  sleep .2
                  send -s -- "$retfil\r"
```

```
                  }

-exact  "    Do you want to send a file? Y/\[N\]: " {
                  sleep .2
                  send -s -- "y\r"
                  expect -exact "    are sending from your system: "
                  sleep .2
                  send -s -- "$retfil\r"
                  }
}

#This script assumes that the TP is registered as a Reporting Agent
#for at least one form type, but that the file being sent is for a form type
#for which the TP is not a Reporting Agent.  Consequently, it responds "n" to the prompt.
expect -exact "    Are you submitting this file as a reporting agent? Y/\[N\]: "
sleep .2
send -s -- "n\r"

#The send_user command writes a message into the TP's local log file
send_user "after send file looking for choice \n "


#When MAIN MENU choice-prompt from EMS is received, send 1 (Logoff).
#After "DISCONNECTING FROM EFS" message is received from EMS, send exit command to telnet
expect {
expect -exact "    Enter your choice: " {
                  sleep .2
                  send -s -- "1\r"
                  send_user "answered 1 to choice\n"
                  expect -exact "DISCONNECTING FROM EFS"
                  send_user "got normal disconnect message \n"
                  exit
                  }
}

#Exit from the script
exit
```